Testes Automatizados Usando Robot Framework – Módulo 1 – Criando o primeiro script com a biblioteca do Selenium

Visão geral

Vamos escrever nosso primeiro script de teste automatizado usando a biblioteca Selenium do Robot Framework. Será um caso de teste bastante simples: vamos abrir o navegador, realizar uma pesquisa no Google e Fechar o navegador. Para tanto cria-se um arquivo com a extensão .robot; define-se as seções obrigatórias dentro do arquivo .robot bem como as bibliotecas e variáveis necessárias para execução do teste.

Arquivo de script de teste

Um script de teste automatizado no Robot Framework é dividido em 4 seções:

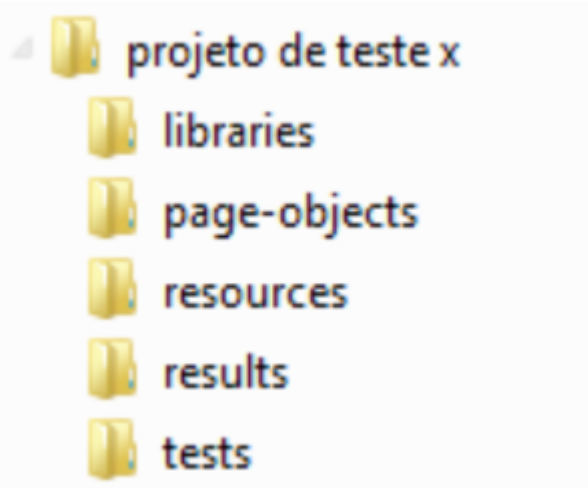
- **Settings**: usada para configuração das bibliotecas utilizadas no teste e inclusão de arquivos auxiliares para o teste como por exemplo Page Objects
- **Variables**: lista de variáveis e seus respectivos valores utilizadas no teste
- **Test Cases**: definição dos cenários/casos de teste
- **Keywords**: definição das palavras-chave utilizadas na implementação dos cenários/casos de teste

A única seção obrigatória é a **Test Cases**, sem ela o script de teste não rodará, as demais seções são opcionais.

Organização dos arquivos do projeto

A organização dos arquivos do projeto é livre, e fica de acordo com a vontade do desenvolvedor definir a melhor forma de distribuir os arquivos com os scripts. Porém aqui deixo um estrutura sugerida de pastas para o projeto de teste:

- **libraries**: bibliotecas adicionais criadas para o projeto
- **page-objects**: definição e mapeamento das telas do sistema sendo testado (vamos ver mais adiante o que é o padrão *page-object*)
- **resources**: arquivos com dados de teste, definições dos ambientes de teste e outros arquivos comuns a todos os testes
- **results**: armazena os relatórios e logs de teste gerados
- **tests**: contém os arquivos com os scripts de teste propriamente ditos



Escrevendo passos de teste

Então vamos lá dar início a criação do nosso primeiro script. Vamos abrir o editor/IDE escolhido e criar um novo arquivo de texto padrão e vamos salvá-lo com

o nome **PrimeiroScript.robot**.

Vamos criar duas seções nesse arquivo. As seções no Robot Framework são definidas com 3 asteriscos como pode ser visto no código abaixo. Temos 2 seções: **Settings** e **Test Cases**. A seção **Test Cases** é obrigatória e é onde o Robot Framework começa a execução. Por padrão os casos de teste são executados na ordem que foram definidos.

```
*** Settings ***
Library                     SeleniumLibrary

*** Test Cases ***
Abrir navegador na página do Google
    Open Browser            http://www.google.com.br   chrome
    Title Should Be         Google
    Input Text              name=q    Robot Framework Selenium Librar
    Press Keys              name=q    ENTER
    Close Browser
```

A seção **Settings** está definindo que vamos usar a biblioteca **SeleniumLibrary** que é responsável por interagir com os navegadores web.

A seção **Test Cases** definimos um nome para o nosso caso de teste. No caso chamamos de **Abrir navegador na página do Google**.

É importante prestar atenção que o Robot Framework é uma linguagem indentada, assim como o Python, e, portanto, os blocos de comando são separados por espaço ou tabulações, formando uma indentação visual obrigatória. Não existem símbolos de “abre” e “fecha” como em outras linguagens. O ideal é usar 4 espaços para a indentação do código e não usar *tabs*.

A primeiro comando que usamos é o **Open Browser** passando 2 parâmetros. O primeiro é a URL que queremos abrir e o segundo informa qual o navegador vamos utilizar. Para mais informações sobre os navegadores suportados pelo Robot Framework consultar a documentação do comando **Open Browser**.

O próximo comando é **Title Should Be**. Esse comando faz uma validação que logo após carregar a página o título deve ser Google.

O terceiro comando é o **Input Text**. Nesse comando passamos 2 parâmetros: o primeiro é o elemento que queremos interagir (que nesse caso é o campo de pesquisa do Google) e o segundo é o texto que vamos escrever nesse elemento.

O quarto comando simulamos o pressionamento de um **ENTER** no campo de pesquisa através do comando **Press Key**.

Por fim fechamos o navegador com o comando **Close Browser**.

Criando Palavras-chave

Podemos melhorar um pouco o teste definindo nossas próprias *keywords* (palavras-chave). Por exemplo, vamos criar uma *keyword* chamada "Abrir navegador" e vamos passar como parâmetro para essa *keyword* uma **URL** e qual navegador queremos abrir. Assim podemos reutilizar futuramente essa *keyword* para outros cenários de teste.

Para escrever nossa *keyword* precisamos declarar ela na seção *Keywords* do nosso arquivo de teste.

```
*** Settings ***
Library                     SeleniumLibrary

*** Test Cases ***
Abrir navegador na página do Google
    Abrir navegador         http://www.google.com.br   chrome
    Title Should Be         Google
    Input Text              name=q    Robot Framework Selenium Librar
    Press Keys              name=q    ENTER
    Close Browser

*** Keywords ***
Abrir Navegador
    [Arguments]   ${URL}    ${navegador}
    Open Browser      ${URL}    ${navegador}
```

O nome da *keyword* que criamos é **Abrir Navegador**. Definimos os parâmetros que seguem a mesma sintaxe da declaração de variáveis através do comando **[Arguments]**.

Uma outra forma mais de passar parâmetros para as *keywords* é declarar os parâmetros diretamente na *keyword*. Eu particularmente prefiro esta maneira. Para tanto, devemos colocar os parâmetros entre aspas, como no exemplo abaixo, **"${URL}"**.

```
*** Settings ***
Library                     SeleniumLibrary

*** Test Cases ***
Abrir navegador na página do Google
    Abrir a página "http://www.google.com.br" com o navegador "chrome
    Title Should Be         Google
    Input Text              name=q    Robot Framework Selenium Librar
    Press Keys              name=q    ENTER
    Close Browser

*** Keywords ***
Abrir a página "${URL}" com o navegador "${navegador}"
    Open Browser      ${URL}    ${navegador}
```

Execução do script automatizado

Para executar o teste devemos abrir uma linha de comando (**DOS Prompt** ou **Powershell**) e executar o seguinte comando:

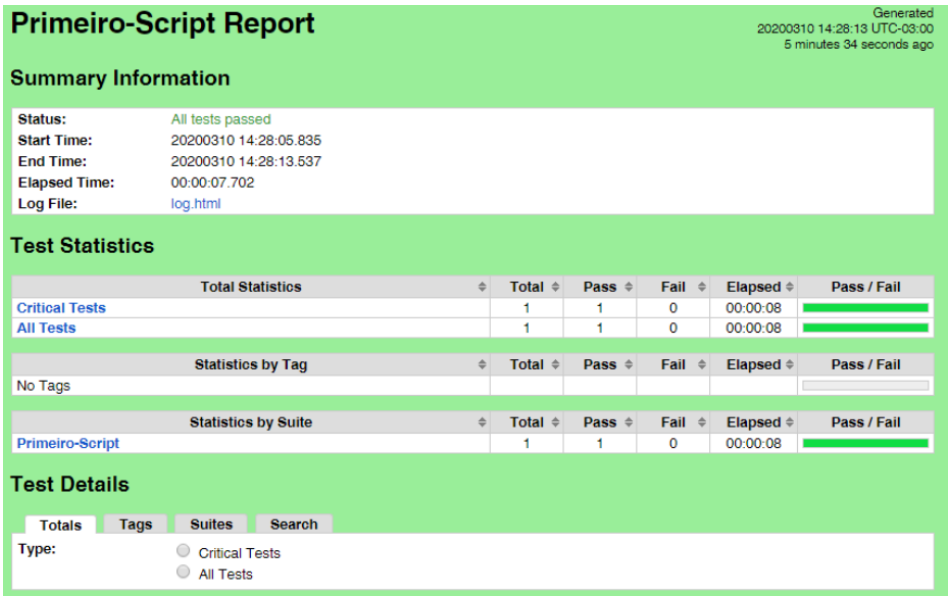
```
robot primeiro-script.robot
==============================================================================
Primeiro-Script
==============================================================================
Abrir navegador na página do Google                                   | PASS |
------------------------------------------------------------------------------
Primeiro-Script                                                       | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
==============================================================================
```

Esse caso de teste não está fazendo nenhuma validação ao final da execução como deveria ser, apenas criamos esse primeiro script para verificar se estamos com tudo

configurado de forma correta e para ter os primeiros passos com o Robot Framework.

Analisando o relatório de testes e log da execução

Ao final da execução o Robot Framework gera um arquivo de relatório e um de log com os resultados da execução.



Primeiro-Script Report

Generated
20200310 14:28:13 UTC-03:00
5 minutes 34 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20200310 14:28:05.835
End Time:	20200310 14:28:13.537
Elapsed Time:	00:00:07.702
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:08	
All Tests	1	1	0	00:00:08	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Primeiro-Script	1	1	0	00:00:08	

Test Details

Totals | Tags | Suites | Search

Type: Critical Tests / All Tests

Rodar em outros navegadores

Podemos rodar esse nosso primeiro script em outro navegador, como por exemplo no Mozilla Firefox. Para tanto devemos ter o Firefox instalado na máquina bem como o *driver*. Basta mudar o parâmetro no comando **Open Browser**, assim o nosso código ficaria assim:

```
*** Settings ***
Library                     SeleniumLibrary

*** Test Cases ***
Abrir navegador na página do Google
   Abrir a página "http://www.google.com.br" com o navegador "firefox
   Title Should Be          Google
   Input Text               name=q    Robot Framework Selenium Library
   Press Keys               name=q    ENTER
   Close Browser
```

```
11
12  *** Keywords ***
13  Abrir a página "${URL}" com o navegador "${navegador}"
14     Open Browser      ${URL}    ${navegador}
```

Like

Be the first to like this.